МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

**«Оптимизация производственных процессов»**

Выполнил: Студент 4 курса

Пекарь Сергей Александрович

Минск 2012

Оглавление

[ВВЕДЕНИЕ 3](#_Toc344252738)

[1. Краткое описание объекта исследования 4](#_Toc344252739)

[2. Описание проблемы 6](#_Toc344252740)

[3. Формализация проблемы 7](#_Toc344252741)

[4. Варианты решения проблемы 9](#_Toc344252742)

[5. Описание выбранного метода решения 12](#_Toc344252743)

[6. Организация сбора информации 16](#_Toc344252744)

[7. Результаты сбора информации и их визуализация 20](#_Toc344252745)

[8. Интерпретация результатов 24](#_Toc344252746)

[9. Оценка результатов 25](#_Toc344252747)

[ВЫВОДЫ 26](#_Toc344252748)

# ВВЕДЕНИЕ

В математике, оптимизации, или математическое программирование, относится к выбору лучших элементов из набора доступных альтернатив.

В простейшем случае это означает решение проблем, в которых одна попытка свести к минимуму или увеличить в реальном функцию путем систематического выборе значения реального или целое с переменными в рамках позволило установить. Это (скалярного реальной стоимостью целевой функции) фактически представляет собой небольшую часть этой области, которая включает в себя большое области прикладной математики. В более общем плане, то это означает нахождение "наилучшие имеющиеся" некоторых значений целевой функции с учетом определенной области, в том числе различные виды объективных функций и различные типы доменов.

# [Краткое описание объекта исследования](#_Toc336801021)

Объектом исследования выступает производственный комплекс утилизации волокнистых отходов. По понятным причинам, в рамках учебного процесса невозможно использовать реальный производственный комплекс. По этой причине реальное производство заменено его математической моделью. Естественно, что, несмотря на то, что комплекс имитирует все функциональные возможности, он никогда не сможет полностью заменить реальное производство. Так как взаимосвязи факторов производства на много более сложные и запутанные, чем это можно представить с помощью математических формул и зависимостей. Но, даже не смотря на эти недостатки, объем задачи огромен: целых 57 переменных, которые включают:

* Экономические факторы;
* природные факторы;
* технологические факторы;
* показатели качества;
* показатели загрязнения
* и многие другие.

И, кроме того, что этих переменных много, они все связаны сложными зависимостями, на изучение которых требуется очень много времени и усилий. Поэтому размерность задачи сокращена до 23-х переменных, которые представляют собой наиболее важные факторы.

Учитывая количество факторов, их можно разделить на группы.

* Экологические:
  + дебет реки;
  + загрязнение водоема.
* Экономические:
  + цена энергии;
  + цена кубического метра речной воды;
  + цена сырья;
  + цена полимерной и волокнистой добавок.

Перечисленные параметры, за исключением загрязнения водоема, являются не зависимыми, так как предприятия не может влиять на их значения. Но кроме зависимых имеются еще 2 вида переменных: зависимые и промежуточные, то есть, только условие задачи поможет определить, к какой группе переменных их отнести. Например, поставив задачу: «Определить степень влияния технологических параметров на качество» промежуточные варианты становятся не зависимыми. А если задача будет звучать так: «Определить условия достижения качества в заданных пределах», то промежуточные и зависимые переменные меняются местами. К зависимым переменным можно отнести:

* прочность выпускаемой продукции;
* пластичность выпускаемой продукции;
* влагопрочность выпускаемой продукции;
* загрязнение водоема.

К промежуточным:

* Расход волокнистой упрочняющей добавки;
* Расход полимерной упрочняющей добавки;
* Степень помола целлюлозы;
* Скорость бумагоделательной машины.

Совокупность данных переменных обусловлена тем, что основной продукцией выбранного комплекса является обойная бумага, а, следовательно, все зависимые и не зависимые переменные отображают процесс производства продукции именно этого типа.

# [Описание проблемы](#_Toc336801022)

При выборе продукции, каждый покупатель всегда обращает внимание на два показателя: качество и цену. И если цена почти линейно зависит от расходов на производство, хранение, транспортировку, налоги и т.д.; то с качеством не все так очевидно, ведь все технологические параметры влияют не на один показатель качества, а на несколько. Причем в разное количественное сочетание факторов дает совершенно разные результаты. Так, например, установив большое значение расхода волокнистой упрочняющей добавки и малое расхода волокнистой упрочняющей добавки, покажет малый прирост прочности продукции. В то время как, установка этих двух параметров на приблизительно равные значения даст существенный прирост или убыль прочности.

В связи с этим, возникает проблема управления качеством производимой продукции, ведь для того, чтобы конкурировать на рынке, нужно поставлять продукцию определенного качества. А для получения такого качества его значения должны поддаваться прогнозированию. Необходимо достаточно точно определять: что случится если изменить значение того или иного технологического параметра, как поведут себя показатели качества, как необходимо будет отрегулировать остальные параметры.

Резюме всему вышесказанному можно подвести, сформулировав следующую задачу: «Получение продукции заданного качества и минимизация загрязнения». То есть будет решаться задача оптимизиции технологического режима для достижения заданных показателей качества продукции.

# [Формализация проблемы](#_Toc336801023)

Для решения задачи, поставленной в предыдущем разделе, а именно: «Выпуск продукции заданного качества и минимизация загрязнения»; необходимо выбрать показатели качества, значения которых, наиболее важны для выпускаемой продукции, а также факторы технологического процесса, оказывающие наиболее сильное влияние на значения показателей качества и показатели загрязнения. Так как на качество продукции, так и на весь технологический процесс в целом огромное влияние оказывает ситуация на рынке: цены на энергию, расходные материалы и т.п. в качестве 2-х независимых переменных были выбраны:

* цена волокнистой упрочняющей добавки;
* цена полимерной упрочняющей добавки.

Так как данная задача предполагает оптимизацию технологического процесса, то в качестве зависимых переменных были выбраны основные факторы технологического процесса, а именно:

* расход волокнистой упрочняющей добавки;
* расход полимерной упрочняющей добавки;
* степень помола целлюлозы;
* скорость бумагоделательной машины.

Так как требуется осуществлять прогнозирование значений показателей качества – второй группой независимых переменных будут являться показатели качества продукции:

* прочность вырабатываемой продукции;
* пластичность вырабатываемой продукции;
* влагопрочность вырабатываемой продукции
* загрязнение водоема.

Для выполнения данной работы были сформулированы следующие требования к качеству продукции:

* прочность выпускаемой продукции: более 10;
* пластичность выпускаемой продукции: более 15;
* влагопрочность выпускаемой продукции: более 15.

Приведенные выше требования являются более высокими по сравнению с теми, которые задаются комплексом по умолчанию.

В качестве фактора загрязнения было выбрано загрязнение водоема, к которому также было установлено требование «не превышать значение в 250 единиц».

# [Варианты решения проблемы](#_Toc336801024)

В процессе проектирования ставится обычно задача определения наилучших, в некотором смысле, структуры или значений параметров объектов. Такая задача называется оптимизационной. Если оптимизация связана с расчетом оптимальных значений параметров при заданной структуре объекта, то она называется параметрической оптимизацией. Задача выбора оптимальной структуры является структурной оптимизацией.

Стандартная математическая задача оптимизации формулируется таким образом. Среди элементов χ, образующих множества Χ, найти такой элемент χ\*, который доставляет минимальное значение f(χ\*) заданной функции f(χ). Для того, чтобы корректно поставить задачу оптимизации необходимо задать:

* Допустимое множество — множество;
* Целевую функцию — отображение;
* Критерий поиска (max или min).

Тогда решить задачу f(x)\to \min_{\vec{x}\in\mathrm{X}} означает одно из:

* Показать, что \mathbb{X}=\varnothing.
* Показать, что целевая функция f(\vec{x}) не ограничена снизу.
* Найти \vec{x}^*\in\mathbb{X}:\;f(\vec{x}^*)=\min_{\vec{x}\in\mathbb{X}}f(\vec{x}).
* Если \nexists \vec{x}^* , то найти \inf_{\vec{x}\in\mathbb{X}}f(\vec{x}).

Общая запись задач оптимизации задаёт большое разнообразие их классов. От класса задачи зависит подбор метода (эффективность её решения). Классификацию задач определяют: целевая функция и допустимая область (задаётся системой неравенств и равенств или более сложным алгоритмом).

Общая запись задач оптимизации задаёт большое разнообразие их классов. От класса задачи зависит подбор метода (эффективность её решения). Классификацию задач определяют: целевая функция и допустимая область (задаётся системой неравенств и равенств или более сложным алгоритмом).

Методы оптимизации классифицируют в соответствии с задачами оптимизации:

* Локальные методы: сходятся к какому-нибудь локальному экстремуму целевой функции. В случае унимодальной целевой функции, этот экстремум единственен, и будет глобальным максимумом/минимумом.
* Глобальные методы: имеют дело с многоэкстремальными целевыми функциями. При глобальном поиске основной задачей является выявление тенденций глобального поведения целевой функции.

Существующие в настоящее время методы поиска можно разбить на три большие группы:

* детерминированные;
* случайные (стохастические);
* комбинированные.

По критерию размерности допустимого множества, методы оптимизации делят на методы одномерной оптимизации и методы многомерной оптимизации.

По виду целевой функции и допустимого множества, задачи оптимизации и методы их решения можно разделить на следующие классы:

Задачи оптимизации, в которых целевая функция f(\vec{x})и ограничения g_i(\vec{x}),\; i=1,\ldots,mявляются линейными функциями, разрешаются так называемыми методами линейного программирования.

В противном случае имеют дело с задачей нелинейного программирования и применяют соответствующие методы. В свою очередь из них выделяют две частные задачи:

* если f(\vec{x})и g_i(\vec{x}),\;i=1,\ldots,m — выпуклые функции, то такую задачу называют задачей выпуклого программирования;
* если \mathbb{X}\subset \mathbb{Z}, то имеют дело с задачей целочисленного (дискретного) программирования.

По требованиям к гладкости и наличию у целевой функции частных производных, их также можно разделить на:

* прямые методы, требующие только вычислений целевой функции в точках приближений;
* методы первого порядка: требуют вычисления первых частных производных функции;
* методы второго порядка: требуют вычисления вторых частных производных, то есть гессиана целевой функции.

Помимо того, оптимизационные методы делятся на следующие группы:

* [аналитические методы](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B0%D0%BB%D0%B8%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D1%8B) (например, [метод множителей Лагранжа](http://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D1%82%D0%B5%D0%BB%D0%B5%D0%B9_%D0%9B%D0%B0%D0%B3%D1%80%D0%B0%D0%BD%D0%B6%D0%B0) и [условия Каруша-Куна-Таккера](http://ru.wikipedia.org/wiki/%D0%A3%D1%81%D0%BB%D0%BE%D0%B2%D0%B8%D1%8F_%D0%9A%D0%B0%D1%80%D1%83%D1%88%D0%B0-%D0%9A%D1%83%D0%BD%D0%B0-%D0%A2%D0%B0%D0%BA%D0%BA%D0%B5%D1%80%D0%B0));
* [численные методы](http://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D1%8B);
* [графические методы](http://ru.wikipedia.org/w/index.php?title=%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D1%8B&action=edit&redlink=1).

В зависимости от природы множества X задачи математического программирования классифицируются как:

* задачи [дискретного программирования](http://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) (или [комбинаторной оптимизации](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%B1%D0%B8%D0%BD%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F_%D0%BE%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F)) — если X [конечно](http://ru.wikipedia.org/wiki/%D0%91%D0%B5%D1%81%D0%BA%D0%BE%D0%BD%D0%B5%D1%87%D0%BD%D0%BE%D1%81%D1%82%D1%8C) или [счётно](http://ru.wikipedia.org/wiki/%D0%A1%D1%87%D1%91%D1%82%D0%BD%D0%BE%D0%B5_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%BE);
* задачи [целочисленного программирования](http://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%BE%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) — если X является [подмножеством](http://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%BE) [множества](http://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%BE) целых чисел;
* задачей нелинейного программирования, если ограничения или [целевая функция](http://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%B5%D0%B2%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) содержат [нелинейные функции](http://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) и X является [подмножеством](http://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%BE) конечномерного [векторного пространства](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE).
* если же все [ограничения](http://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%B3%D1%80%D0%B0%D0%BD%D0%B8%D1%87%D0%B5%D0%BD%D0%B8%D0%B5&action=edit&redlink=1) и [целевая функция](http://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%B5%D0%B2%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) содержат лишь линейные функции, то это — задача линейного программирования.

Кроме того, разделами математического программирования являются [параметрическое программирование](http://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [динамическое программирование](http://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и [стохастическое программирование](http://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%BE%D1%85%D0%B0%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5).

Математическое программирование используется при решении оптимизационных задач [исследования операций](http://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B9).

Способ нахождения экстремума полностью определяется классом задачи. Но перед тем, как получить математическую модель, нужно выполнить 4 этапа моделирования:

* Определение границ системы оптимизации
* Отбрасываем те связи объекта оптимизации с внешним миром, которые не могут сильно повлиять на результат оптимизации, а, точнее, те, без которых решение упрощается
* Выбор управляемых переменных
* «Замораживаем» значения некоторых переменных (неуправляемые переменные). Другие оставляем принимать любые значения из области допустимых решений (управляемые переменные)
* Определение ограничений на управляемые переменные
* … (равенства и/или неравенства)
* Выбор числового критерия оптимизации (например, [показателя эффективности](http://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BA%D0%B0%D0%B7%D0%B0%D1%82%D0%B5%D0%BB%D1%8C_%D1%8D%D1%84%D1%84%D0%B5%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D0%B8))
* Создаём целевую функцию

## Описание выбранного метода решения

Для выполнения данной лабораторной работы будет использоваться метод случайного локального поиска предоставляемый программой Optim.

Во всех регулярных итеративных методах поиска минимума http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image001.gif для получения текущей оценки http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image002.gif параметра http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image003.gif делается неслучайный шаг, однозначно определяемый либо значением градиента http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image004.gif при http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image005.gif, либо значением самой функции http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image001.gif.

В методах случайного поиска при переходе от http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image006.gif к http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image002.gif делается случайный шаг http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image007.gif, где http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image008.gif - единичный случайный вектор, чаще всего равномерно распределенный в http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image009.gif-мерной единичной сфере; http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image010.gif — величина шага. В этом случае

http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image011.gif

Существуют различные модификации алгоритма (2.45). В этом алгоритме случайный шаг делается только в том случае, если http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image012.gif, в противном случае система остается в предыдущем состоянии http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image013.gif. В других алгоритмах, например, «с наказанием случайностью» случайный шаг делается только тогда, когда предыдущий шаг был неудачным, и т. д.

Наконец, если в регулярном градиентном алгоритме величину http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image010.gif сделать случайной, то мы также получим алгоритм случайного поиска:

http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image014.gif, где http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image015.gif — реализация случайной величины http://www.sernam.ru/archive/arch.php?path=../htm/book_zip/files.book&file=zip_25.files/image016.gif.

Так же программа Optim предоставляет нам для оптимизации метод скользящего допуска.

[Метод скользящего допуска](javascript:termInfo(%22Метод%20скользящего%20допуска%22)) существенно использует множество

|  |  |
| --- | --- |
| http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=6/?k=10 |  |

где неотрицательный скаляр http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=7/?k=10— [критерий скользящего допуска](javascript:termInfo(%22Критерий%20скользящего%20допуска%22)), http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=8/?k=10— неотрицательно определенный функционал над множеством всех [ограничивающих функций](javascript:termInfo(%22ограничивающих%20функций%22)) http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=9/?k=10, http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=10/?k=10.

При этом функционал http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=8/?k=10должен быть сконструирован таким образом, чтобы http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=11/?k=10при http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=12/?k=10и значение http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=8/?k=10возрастало по мере удаления точки http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=13/?k=10от границы [области допустимых значений](javascript:termInfo(%22области%20допустимых%20значений%22)) http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=2/?k=10. [Критерий скользящего допуска](javascript:termInfo(%22Критерий%20скользящего%20допуска%22)) http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=7/?k=10определяет требуемую точность выполнения ограничений, которые формируют область допустимых значений http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=2/?k=10, и конструируется таким образом, чтобы обеспечить его уменьшение с ростом количества итераций http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=14/?k=10.

Точка http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=13/?k=10называется [допустимой точкой](javascript:termInfo(%22Допустимая%20точка%22)), если http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=11/?k=10, [почти допустимой точкой](javascript:termInfo(%22Почти%20допустимая%20точка%22)) — если http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=15/?k=10, [недопустимой точкой](javascript:termInfo(%22Недопустимая%20точка%22)) — если http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=16/?k=10. Поскольку величина http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=7/?k=10с ростом номера итерации уменьшается, отклонение от границы области http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=2/?k=10, при котором точка считается допустимой, сужается, так что в пределе рассматриваются только допустимые точки.

[Метод скользящего допуска](javascript:termInfo(%22Метод%20скользящего%20допуска%22)) может быть скомбинирован со многими из рассмотренных ранее многомерных методов локальной безусловной оптимизации. Будем называть метод, с которым комбинируется метод скользящего допуска, [базовым методом](javascript:termInfo(%22Базовый%20метод%22)).

Одна итерация [метода скользящего допуска](javascript:termInfo(%22метода%20скользящего%20допуска%22)) состоит из одного или двух этапов:

1) С помощью [базового метода](javascript:termInfo(%22базового%20метода%22)), исходя из сточки http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=17/?k=10, выполняем итерацию по решению [задачи локальной безусловной оптимизации](javascript:termInfo(%22задачи%20локальной%20безусловной%20оптимизации%22))

|  |  |
| --- | --- |
| http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=18/?k=10 |  |

находим точку http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=19/?k=10. Если http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=20/?k=10(точка http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=19/?k=10является [допустимой точкой](javascript:termInfo(%22допустимой%20точкой%22)) или [почти допустимой точкой](javascript:termInfo(%22почти%20допустимой%20точкой%22))), то полагаем http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=21/?k=10и заканчиваем данную итерацию.

2) Если http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=22/?k=10(точка http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=19/?k=10является недопустимой), то отыскиваем точку http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=23/?k=10, лежащую ближе к границе области http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=2/?k=10. Для этого с помощью того же [базового метода](javascript:termInfo(%22базового%20метода%22)), исходя из точки http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=19/?k=10, решаем [задачу локальной безусловной оптимизации](javascript:termInfo(%22задачу%20локальной%20безусловной%20оптимизации%22))

|  |  |
| --- | --- |
| http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=24/?k=10 |  |

с условием окончания итераций

|  |  |
| --- | --- |
| http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=25/?k=10 |  |

и заканчиваем данную итерацию.

Достоинством [метода скользящего допуска](javascript:termInfo(%22метода%20скользящего%20допуска%22)) является то, что степень нарушения ограничений по мере приближения к минимуму минимизируемой функции постепенно уменьшается. Т.е. на первых итерациях ограничения могут удовлетворяться приближенно, а высокая точность удовлетворения ограничений необходима лишь в окрестности решения. Это обстоятельство позволяет сократить полный объем вычислений по сравнению с другими методами.

Одна из сложностей применения [метода скользящего допуска](javascript:termInfo(%22метода%20скользящего%20допуска%22)) — возможные осцилляция решения относительно границы области http://bigor.bmstu.ru/?frm/?doc=MO/ch0902.mod/?n=2/?k=10.

# [Организация сбора информации](#_Toc336801026)

Для сбора информации в данной лабораторной работе используются модели, сделанные в предыдущей работе «Моделирование производственных процессов».

В первой вкладке «Независимые переменные» вводится некоторое количество переменных, которым затем присваиваются имена и задаются верхняя и нижняя границы (рисунок 6.1).

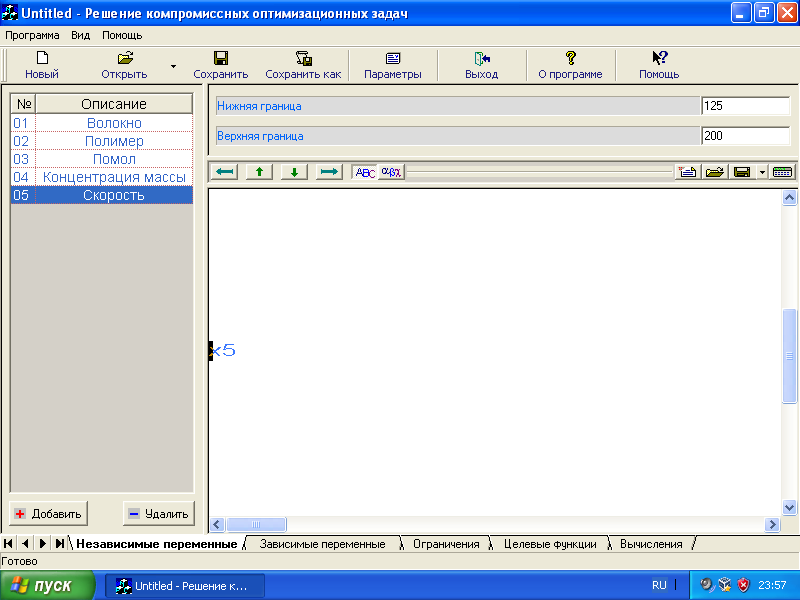


Рисунок 6.1 – Ввод независимых переменных и их границ

В данной работе, как и в предыдущей, заданы следующие ограничения:

* Волокно: 125-200;
* Полимер: 75-150;
* Помол: 20-30;
* Концентрация массы: 0.5-0.9;
* Скорость: 125-150.

Выбор данных диапазонов обусловлен тем, что технологический режим, основанный на значениях факторов в данных диапазонах, обуславливал наиболее удовлетворительные результаты.

На следующей вкладке «Независимые переменные» (рисунок 6.2) задаются независимые переменные, их имена и формулы, по которым они вычисляются. Данными переменными являются те, по которым были построены модели в предыдущих лабораторных работах, формулы для вычисления данных переменных также основаны на этих моделях.

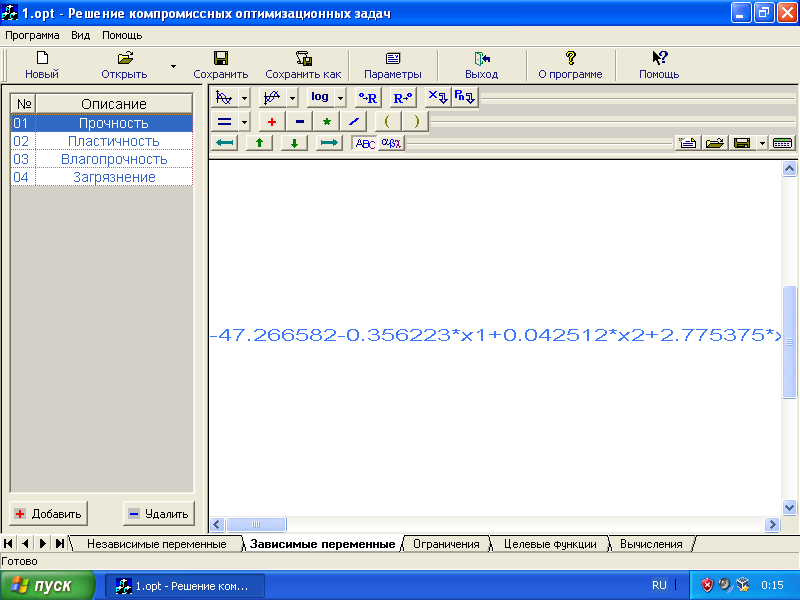


Рисунок 6.2 – Ввод зависимых переменных

Следующим действием является задание ограничений для каждой независимой переменной (рис. 6.3). То есть, по сути формируются требования к результатам. Для каждой зависимой переменной были сформулированы следующие требования: прочность более 10, пластичность и влагопрочность более 15, загрязнение менее 250. При данных требованиях будет обеспечено высокое качество, но при этом требования к загрязнению позволят нанести наименьший вред природе.

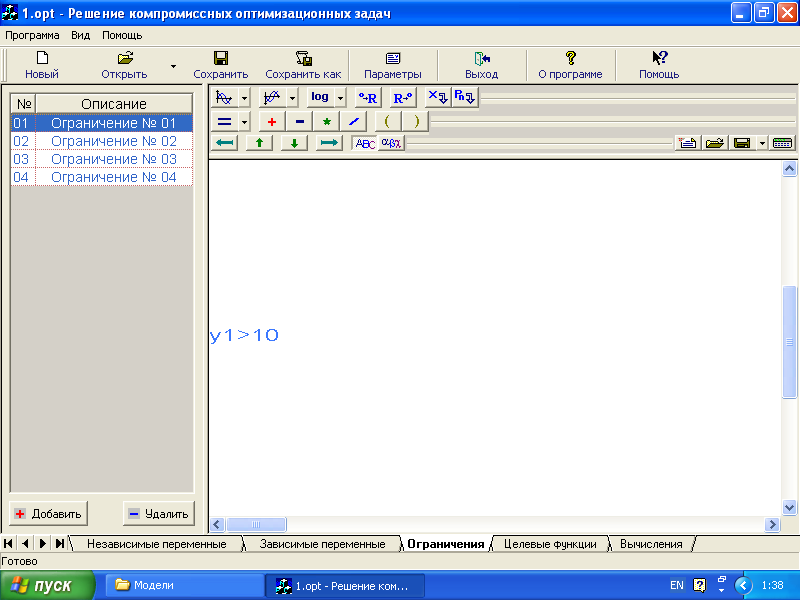
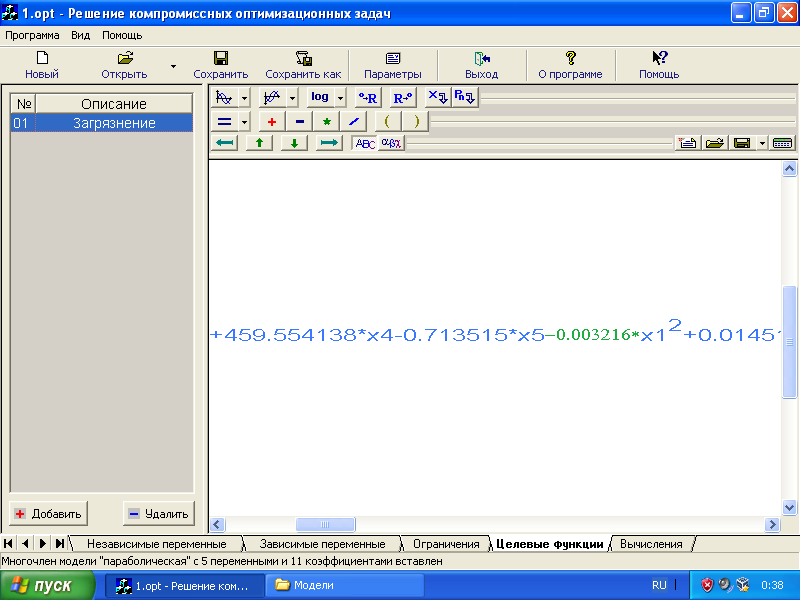


Рисунок 6.3 – Ввод зависимых ограничений

На вкладке «Целевые функции» (рис. 6.4) необходимо задать функцию цели для оптимизации. Так как обеспечить необходимое качество в данных диапазонах не составит большого труда, следует обратить внимание на побочный эффект роста качества продукции, а именно – загрязнение, ведь оно увеличивается с увеличением показателей технологического режима, необходимых для получения заданного качества.

Так как данные по загрязнению и остальным параметрам были собраны в предыдущей работе, и по ним построены модели, причем некоторые из них являются адекватными, будет целесообразно использовать уже построенную модель загрязнения.

Рисунок 6.3 – Ввод функции цели

На вкладке вычисления можно получить результаты решения оптимизационной задачи, что и должно быть проделано в качестве следующего шага.

# Результаты сбора информации и их визуализация

Вкладка «Вычисления» предоставляет 2 метода решения: метод случайного локально поиска и метод скользящего допуска.

Метод случайного локального поиска очень эффективен по той причине, что в решение вносится элемент случайности, что уменьшает шанс попадания в неверную область, но обратная сторона медали в том, что для нахождения решения необходимо применение данного метода несколько раз.

Первый опыт осуществляется с шагом 0.01 (рис. 7.1).

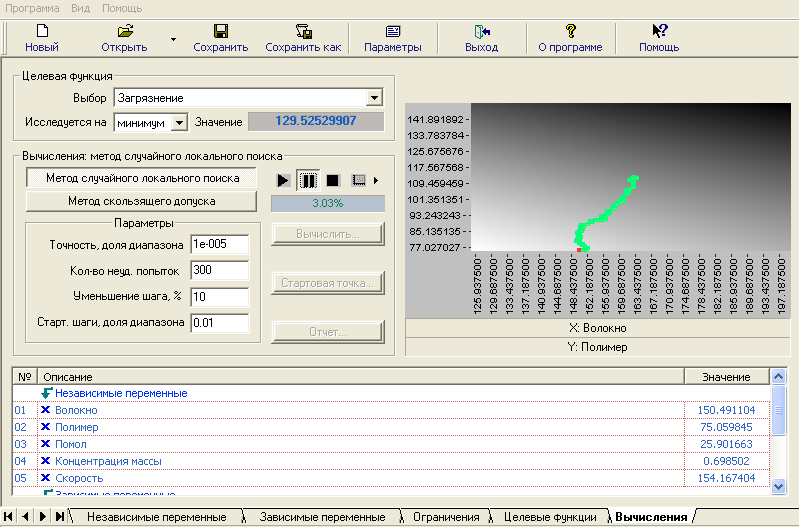


Рисунок 7.1 – Результат вычислений при шаге 0.01

Как видно, из данного рисунка удалось получить загрязнение равное 129.

Следующий опыт осуществляется с шагом 0.05 (рис. 7.2)

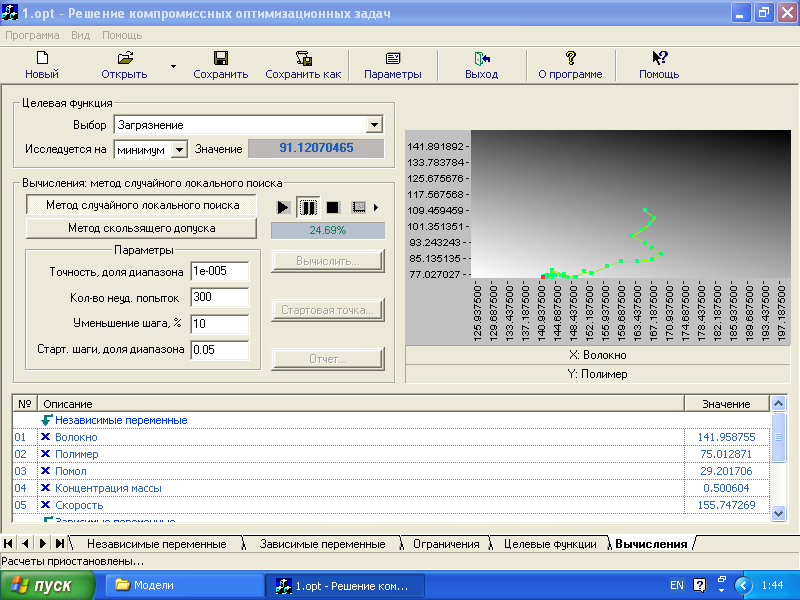


Рисунок 7.2 – Результат вычислений при шаге 0.05

Как видно из рисунка, увеличение шага позволило снизить загрязнение почти на 40 единиц, что говорит о том, что увеличение шага позволило выйти из области, которая ранее считалась оптимальной.

Далее увеличение шага до 0.1 (рис. 7.3)

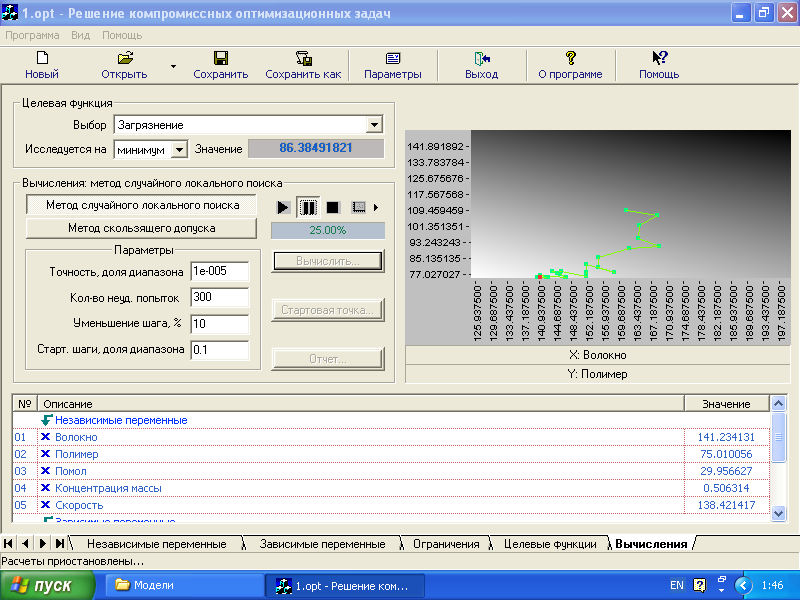


Рисунок 7.3 – Результат вычислений при шаге 0.1

Как видно, замечена тенденция получения более оптимального решения с увеличением шага.

Далее шаг 0.5 (рис. 7.4)

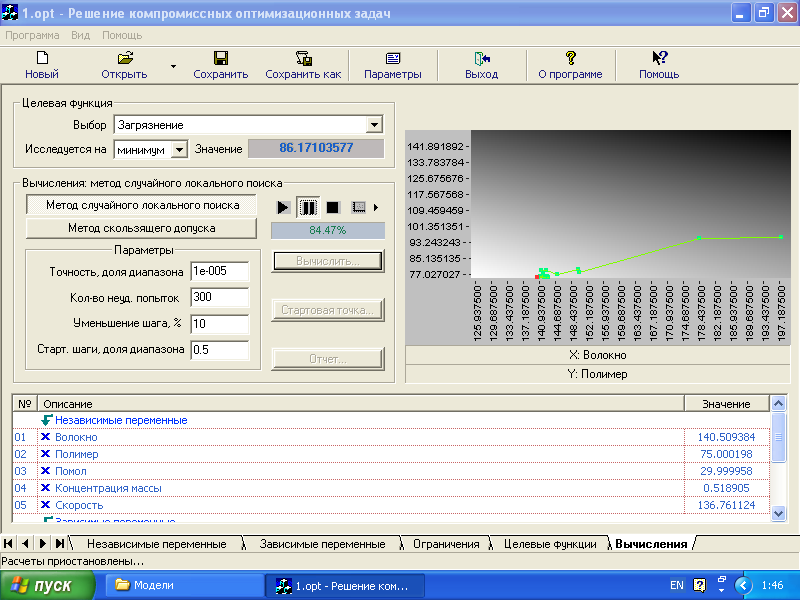


Рисунок 7.4 – Результат вычислений при шаге 0.5

Как видно из рисунка, результаты поиска почти не изменились.

И последний шаг 0,99 (рис. 7.5)

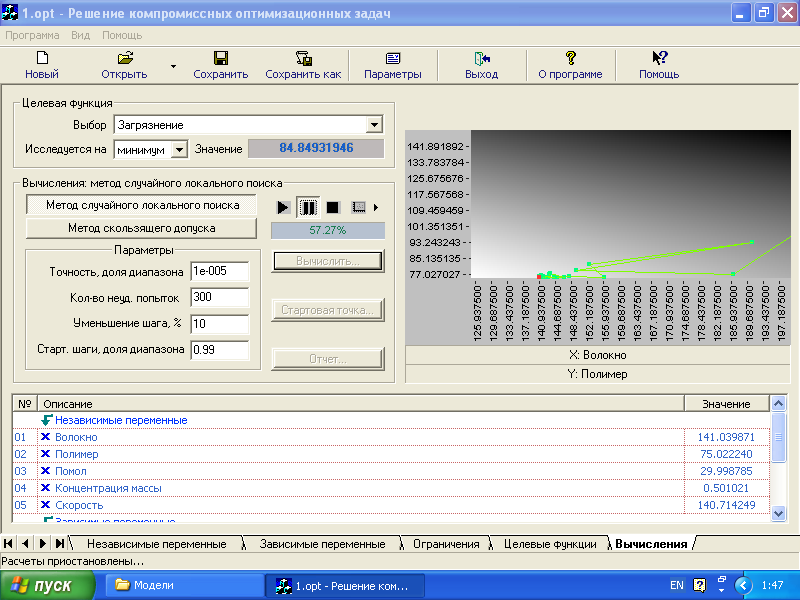


Рисунок 7.5 – Результат вычислений при шаге 0.99

Как видно, было получено самое маленькое значение.

Также был выполнен поиск решения методом скользящего допуска (рис. 7.6).

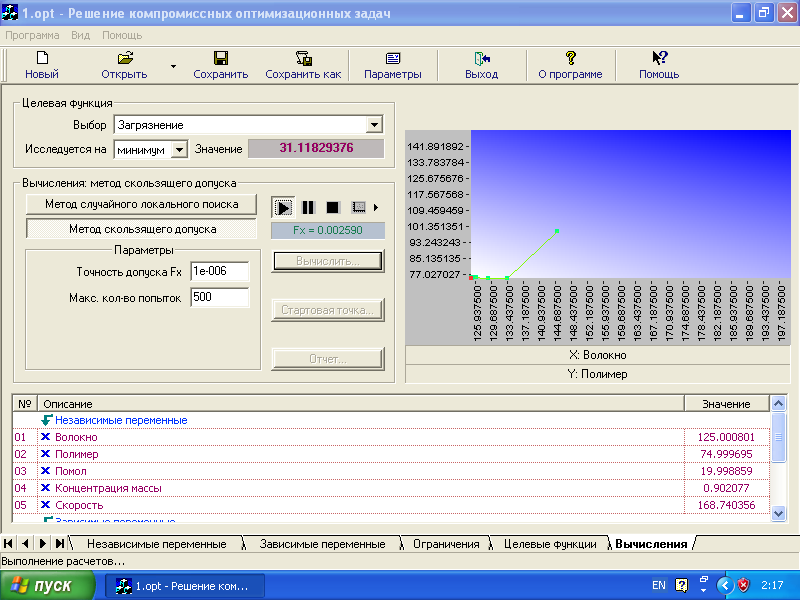


Рисунок 7.5 – Результат вычислений методом скользящего допуска

Как видно из рисунка 7.5 - метод скользящего допуска по результатам решения обошел метод случайного локального поиска со всеми значениями шага.

# [Интерпретация результатов](#_Toc336801029)

По результатам опыта с минимальным шагом можно увидеть, что было получено загрязнение 129,525. По риуснку с траекторией движения 7.1 видно, что была пройдена далеко не вся область допустимых значений, что говорит, о том, что маленький шаг допустим только при возможности запоминания большого количества точек.

Результаты опыта с шагом 0.05 дали результат 91,12, что говорит о том, что Увеличение шага без изменения количества точек позволяет исследовать более высокую область, а следовательно и получить лучшее решение.

Опыт с шагом 0.1 дал результат 86,384, что доказывает необходимость увеличения шага на больших площадях допустимых решений.

Шаг 0.5 дал еще лучший результат, но учитывая элемент случайности можно сказать, что увеличение шага до 0.5 не приводит к существенному изменению результатов.

Опыт с максимальным шагом позволил получить наилучшее значение, но при этом, прогон опыта несколько раз показывал не только худшие, но также и недопустимые результаты (рис. 8.1), что говорит, об опасности применения такого значения шага.

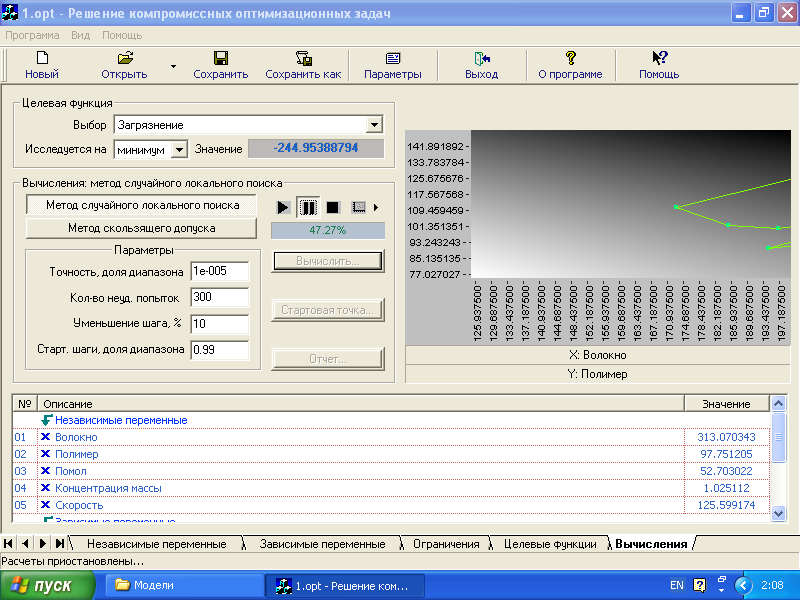


Рисунок 8.1 – Отображение некорректных результатов при большом шаге

# [Оценка результатов](#_Toc336801030)

По результатом работы видно, что при небольшом количестве запоминаемых точек (в данном случае 100), алгоритм случайного локального поиска позволяет получать лучшие результаты при большем значении шага. Однако, для получения такого решения вычислительный эксперимент должен выполняться несколько раз, причем при одном и том же значении шага. Кроме того, следует обратить внимание на тот факт, что несмотря на возможность выхода алгоритма и «ямы псевдо оптимальных решений» при увеличении шага – теряется его точность, что доказано на примере работы алгоритма при задании максимального шага (рис. 8.1).

Наилучшие результаты показал метод скользящего допуска, но также как и метод случайного локально поиска он подвержен ошибкам реализации, кроме того, для проведения нескольких вычислительных экспериментов с большим количеством попыток необходимо огромное количество вычислительных ресурсов и времени.

Исходя из вышесказанного, можно сделать следующие выводы:

* Метод случайного локального поиска позволяет решить оптимизационную задачу за небольшое время, но при этом точность метода зависит от количества запоминаемых точек и величины шага, кроме того, имеется вероятность ошибки при решении задачи данным методом при выполнении вычислительного эксперимента малое число раз и при большом шаге.
* Метод скользящего допуска позволяет получить лучшее решение нежели метод случайного локального поиска, но время выполнения одного вычислительного эксперимента крайне велико и требует большого количества вычислительных ресурсов, кроме того этот метод также подвержен ошибкам.

# [ВЫВОДЫ](#_Toc336801031)

В данной работе были освоены методы решения оптимизационных задач при помощи программного средства «Решение компромиссных оптимизационных задач».

Оптимизация производственного процесс представляет собой основную задачу для любого человека, управляющего предприятием. И применение данных методов делает этот процесс легким в понимании, интерпретации, так как из всех методов системного анализа, данный метод показал себя как наиболее понятный с точки зрения формулирования рекомендации по заданию технологического режима производства.

В сочетании с моделированием производственных процессов, решение оптимизационных задач является наиболее мощным и результативным методом системного анализа из всех изученных ранее.